

WHAT IS CLAIMED IS:

- 1 1. A method for selectively deploying enterprise software comprising:
2 for each deployable software component in an preselected input archive file,
3 comparing interfaces for the deployable software component identified in a first
4 descriptor file in said input archive file and a second descriptor file in a preselected
5 output archive file;
6 if the comparing step miscompares for a first deployable software component,
7 tagging said first deployable software component;
8 if the comparing step miscompares for a second deployable software
9 component, tagging said second deployable software component; and
10 deploying each tagged deployable software component.
- 1 2. The method of claim 1 wherein tagging a deployable software
2 component comprises storing a name of the deployable software component in a file.
- 1 3. The method of claim 1 further comprising:
2 if the first descriptor file and second descriptor file compare for the first
3 deployable software component, comparing a size of a binary class file for the first
4 deployable software component in the input and output archive files; and
5 if the size of said binary class files miscompare, tagging the first deployable
6 software component.

1 4. The method of claim 1 further comprising:
2 if the first descriptor file and second descriptor file compare for the first
3 deployable software component, introspecting a binary class file for the first
4 deployable software component in the input and output archive files; and
5 if, in response to the introspection, a signature or return type of an interface of
6 said binary class files miscompare, tagging the first deployable software component.

1 5. The method of claim 1 further comprising:
2 opening said preselected output archive file; and
3 if the step of opening the preselected output archive fails, tagging each
4 deployable software component in the input archive file.

1 6. The method of claim 5 wherein the step of tagging each deployable
2 software component is performed in response to the step of opening the preselected
3 output archive throwing an exception.

1 7. The method of claim 1 wherein the comparing, tagging and deploying
2 steps are performed in response to an execution of a build script invoking a selective
3 deployer utility.

1 8. A computer program product embodied in a machine-readable medium
2 for selectively deploying enterprise software, the program product comprising
3 programming instructions for:

4 for each deployable software component in an preselected input archive file,
5 comparing interfaces for the deployable software component identified in a first
6 descriptor file in said input archive file and a second descriptor file in a preselected
7 output archive file;

8 if the comparing operation mismatches for a first deployable software
9 component, tagging said first deployable software component;

10 if the comparing operation mismatches for a second deployable software
11 component, tagging said second deployable software component; and

12 deploying each tagged deployable software component.

1 9. The program product of claim 8 wherein tagging a deployable
2 software component comprises storing a name of the deployable software component
3 in a file.

1 10. The program product of claim 8 further comprising programming
2 instructions for:

3 if the first descriptor file and second descriptor file compare for the first
4 deployable software component, comparing a size of a binary class file for the first
5 deployable software component in the input and output archive files; and

6 if the size of said binary class files mismatch, tagging the first deployable
7 software component.

1 11. The program product of claim 8 further comprising programming
2 instructions for:

3 if the first descriptor file and second descriptor file compare for the first
4 deployable software component, introspecting a binary class file for the first
5 deployable software component in the input and output archive files; and

6 if, in response to the introspection, a signature or return type of an interface of
7 said binary class files miscompare, tagging the first deployable software component.

1 12. The program product of claim 8 further comprising programming
2 instructions for:

3 opening said preselected output archive file; and

4 if the operation of opening the preselected output archive fails, tagging each
5 deployable software component in the input archive file.

1 13. The program product of claim 12 wherein the operation of tagging
2 each deployable software component is performed in response to the operation of
3 opening the preselected output archive throwing an exception.

1 14. The program product of claim 8 wherein the comparing, tagging and
2 deploying operations are performed in response to an execution of a build script
3 invoking a selective deployer utility.

1 15. A data processing system for selectively deploying enterprise software
2 comprising:

3 circuitry operable for, for each deployable software component in an
4 preselected input archive file, comparing interfaces for the deployable software
5 component identified in a first descriptor file in said input archive file and a second
6 descriptor file in a preselected output archive file;

7 circuitry operable for, if the comparing operation miscompares for a first
8 deployable software component, tagging said first deployable software component;

9 circuitry operable for, if the comparing operation miscompares for a second
10 deployable software component, tagging said second deployable software component;
11 and

12 circuitry operable for deploying each tagged deployable software component.

1 16. The data processing system of claim 15 wherein tagging a deployable
2 software component comprises storing a name of the deployable software component
3 in a file.

1 17. The data processing system of claim 15 further comprising:
2 circuitry operable for, if the first descriptor file and second descriptor file
3 compare for the first deployable software component, comparing a size of a binary
4 class file for the first deployable software component in the input and output archive
5 files; and

6 circuitry operable for, if the size of said binary class files miscompare, tagging
7 the first deployable software component.

1 18. The data processing system of claim 15 further comprising
2 programming instructions for:

3 circuitry operable for, if the first descriptor file and second descriptor file
4 compare for the first deployable software component, introspecting a binary class file
5 for the first deployable software component in the input and output archive files; and

6 circuitry operable for, if, in response to the introspection, a signature or return
7 type of an interface of said binary class files miscompare, tagging the first deployable
8 software component.

1 19. The data processing system of claim 15 further comprising
2 circuitry operable for opening said preselected output archive file; and

3 circuitry operable for, if the operation of opening the preselected output
4 archive fails, tagging each deployable software component in the input archive file.

1 20. The data processing system of claim 19 wherein the operation of
2 tagging each deployable software component is performed in response to the
3 operation of opening the preselected output archive throwing an exception.